X-562-68-387

PREPRINT

NASA TM X- 63388

A GENERAL-PURPOSE ON-BOARD PROCESSOR FOR SPACECRAFT APPLICATION

OCTOBER 1968

GSFC

GODDARD SPACE FLIGHT CENTER

GREENBELT, MARYLAND

X-562-68-387

A GENERAL-PURPOSE ON-BOARD PROCESSOR FOR SPACECRAFT APPLICATION

M. Tharpe
C. Trevathan
R. Hartenstein
J. Novello

Space Electronics Branch

Information Processing Division

October 1968

GODDARD SPACE FLIGHT CENTER
Greenbelt, Maryland

FOREWORD

This document serves to update the information initially published in X-562-67-202, "A GENERAL-PURPOSE ON BOARD PROCESSOR FOR SCIENTIFIC SPACECRAFT," dated July 1967. Generally, this document discusses the hardware aspects of the computer system. For readers interested in the support software system, reference is made to X-562-68-388, "SUPPORT SOFTWARE FOR THE SPACE ELECTRONICS BRANCH ON-BOARD PROCESSOR."

A GENERAL-PURPOSE ON-BOARD PROCESSOR FOR SPACECRAFT APPLICATION

R. Hartenstein, J. Novello, M. Tharpe, and C. Trevathan

Space Electronics Branch

Information Processing Division

ABSTRACT

This paper describes a general-purpose stored program digital computer, being developed by the Space Electronics Branch for use on scientific spacecraft, and its laboratory configuration. The computer, called the On-Board Processor (OBP), has been designed to meet the objectives of multimission applicability, increased experiment capability, and simplified software construction with reliability being emphasized at both the system and component levels. This report includes a description of the processor's grammatically structured machine language together with a delineation of the central processing unit, input/output philosophy, and modular memory. Circuitry and packaging techniques for both the engineering model and flight systems are presented. The laboratory configuration interconnects the OBP with an SDS 920 computer providing the capability of spacecraft computer and environment simulation.

CONTENTS

CONTENTS (Continued)

ILLUSTRATIONS

TABLES

INTRODUCTION

Traditionally, spacecraft data-handling systems have performed a minimum of on-board data-processing and have returned all experimental data for analysis by the experimenter. Space scientists have been reluctant to discard data before transmittal to the ground on the basis of criteria established before launch and have been unable, for the most part, to adjust their information-gathering apparatus to meet the situations encountered without compromising reliability with complex flight hardware. At present, many arguments exist for developing more extensive on-board data-processing techniques. The advent of microelectronics and the expense of ground-processing the enormous volume of data being returned from space have given considerable impetus to the idea of "flying" systems capable of reducing the data to be transmitted to "information" in the true sense of the word. Further, as various phenomena are investigated in more and more detail, increasingly more discriminating measurements must be transmitted within the bandwidth constraints imposed by spacecraft communications systems. The use of such devices as logarithmic counters by individual experiments has aided this data-transmitting effort to some extent, but the fact remains that the capacity of present spacecraft communications systems has been reached.

To circumvent this "information return impasse," the Space Electronics Branch is developing an on-board data-processing system in the form of a computer which will service individual experiment and spacecraft sensors to provide a more efficient data stream through data compression and the optimization of the data format. Since the computer can be reprogrammed and has a large computational capacity, any degree of on-board processing can be performed by reprogramming from the ground after launch. In this way a spacecraft can be launched with a simple data format and, as the characteristics of the phenomena and the behavior of the instruments in orbit are ascertained, additional degrees of data processing can be implemented.

This computer, called the On-Board Processor (OBP), will be able to control the sampling and formatting of telemetry data, provide data storage, and perform lengthy and sophisticated computations for use in processing experiment, attitude control, and other spacecraft signals. The OBP memory will also provide storage for commands to be acted upon after specified time intervals or upon certain inputs from the spacecraft subsystems and/or experiments. In addition, by applying this OBP to prelaunch and postlaunch checkout of the spacecraft and experiments the effectiveness of those phases of spacecraft operations should greatly increase.

GENERAL DESCRIPTION

The On-Board Processor has been developed specifically for spacecraft application. The system is modular and consists, in a minimum configuration, of one central processor unit (CPU), one memory unit, and one input/output (I/O) unit, all connected to a common and redundant data bus. The interface circuits will be designed so that the data bus will be impervious to a significantly probable single component failure in an input or output circuit.

A CPU is composed of 1300 microcircuits and will dissipate 5 watts of power. The unit is organized as an 18-bit parallel binary machine with 6.25 usec add time, one full-length index register, and hardware multiply/divide. Each memory module will have 4,096 18-bit words. Power dissipation by the core memory is proportional to usage, with close to 30 watts dissipated for continuous access and less than 100 milliwatts required for standby operation. The addition of up to 16 memory modules, giving a storage capacity of 65,536 18-bit words, requires very little additional power. The complexity and size of the I/O unit will depend largely upon mission requirements because an I/O unit will be specifically tailored for each mission. However, an I/O unit capable of accommodating most missions should not exceed the CPU in size or power requirements.

The flight memory module will measure 8.0 by 4.5 by 3.7 inches. The flight I/O and CPU will each have approximate dimensions of 8.0 by 6.5 by 4.5 inches. Thus a minimum system of one CPU, one 4K memory unit, and one I/O unit would occupy about 600 cubic inches and would dissipate from 10 to 40 watts of power, depending upon memory usage. The capability and reliability of the system can be expanded by adding additional memory as well as a standby CPU to be switched into service upon failure in the first CPU. It is significant that the future development and implementation of Non Destructive Readout (NDRO) plated wire memories will yield an operating power reduction of 50 to 60 percent.

The Space Electronics Branch has the overall responsibility for the Processor development. An engineering model of the CPU was designed and fabricated by Westinghouse Defense and Space Center, Aerospace Division under Goddard contract. A commercial Honeywell memory was used in the engineering model. The Space Electronics Branch developed a general I/O, and also provided the capability for ground checkout and experimentation by interfacing the OBP system with an SDS-920 computer to form a laboratory configuration. In this configuration, interface between the OBP and the outside world is provided by the 920 peripheral equipment which communicates with the OBP memory via simulated command and telemetry links.

OBJECTIVES

Design

In developing the On-Board Processor, several basic goals have been set. One major objective is to ensure that the OBP will have mutli-mission applicability. In other words, it will not be designed for a specific mission but rather with the ability to adapt to large or small missions. This requirement will be met by defining the I/O interface between the CPU and the I/O box, and then tailoring the box for a specific mission. The CPU will have high-speed computational capability so that lengthy operations such as statistical analyses will be possible. Thus, through the use of such on-board data-processing, a second major objective, that of increased experiment capability, can be met.

A third major design objective is the development of a software system which will, by its simplicity, allow efficient program construction and debugging by the user. To facilitate programming and to enable the program to be somewhat self-documenting, the basic machine language is grammatically structured restricted English. The instruction set is such that software simulators can be easily implemented on existing general-purpose stored-program computers. A comprehensive executive system will maintain control over subprograms, maintain interrupt priority, and automatically check, through a real-time clock interrupt, to ensure recovery from problem areas such as infinite loop execution.

Reliability

Ultra-reliability is a necessary attribute for a system of this sort which, for many applications, will be operating in an environment untenable by man. Reliability is being emphasized in the design of the Processor at both the system and component levels.

At the system level, the modular organization enables reliability to be enhanced by the ease with which redundant functional subsystems may be interconnected. Figure 1 is a configuration of multiple memory, central processor, and I/O units. The interconnecting address and I/O data bus are totally redundant, including all line drivers and receivers. For such a system, any number of spare modules may be normally connected and powered ON by ground command as failures occur in operating modules. For example, switching power from a malfunctioning memory to a spare, preloaded with the operational program, would result in 100 percent system repair.

Steps taken to assure maximum reliability at the component level are:

- Maximum use is made of monolithic integrated circuits.

- All circuit components will operate well below rated values to minimize failures due to electrical stresses.

- Only tried and proven, reliable, circuit techniques and components will be used.

- All active elements will be burned in to detect early failures and to minimize drift due to aging.

- The number of electrical connections will be minimized with all internal connections either welded or soldered.

- All electronic subassemblies will be encapsulated.

To further enhance the reliability of the CPU and I/O subsystems, future implementations will include some degree of circuit redundancy. A study is currently underway to determine the optimum interconnection of redundant logic within the units. The concept to be employed involves the application of spare functional circuits which would normally be unpowered and switched ON only as needed for repair. An example of how the spare circuits may be implemented in the CPU is presented.

All registers and other bit-oriented logic will be partitioned into functional columns of three bits each. For an 18-bit organization, the CPU will contain six of these columns for operation in addition to one or two columns for spares. Each spare will be interconnected through input and output gating such that it could function as any one of the six active columns. Thus, a malfunction caused by a faulty column could be repaired by a control signal which would turn OFF the bad circuit and switch its inputs and outputs to a spare column. The control signal would be the result of a ground generated command. This implementation does not imply an adaptive form of self-repair.

The remaining circuits in the CPU are used as miscellaneous control and instruction execution logic and are not symmetrically interconnected. Here, the logic circuits will be partitioned into critical and noncritical sections with the critical section being paralleled by an unpowered spare. A failure in the noncritical section cannot be repaired and may cause a slight degradation in performance. A failure in the critical section can be repaired by switching power from the operating to the standby section. Again, this switching would be controlled by a command issued from the ground. These commands required for repair can be generated only after the failure has been detected and analyzed. This process will be accomplished by executing diagnostic routines on board and telemetering the results to control center operators.

Preliminary study indicates that a single spare 3-bit column and a single spare critical control section when compared with a non redundant

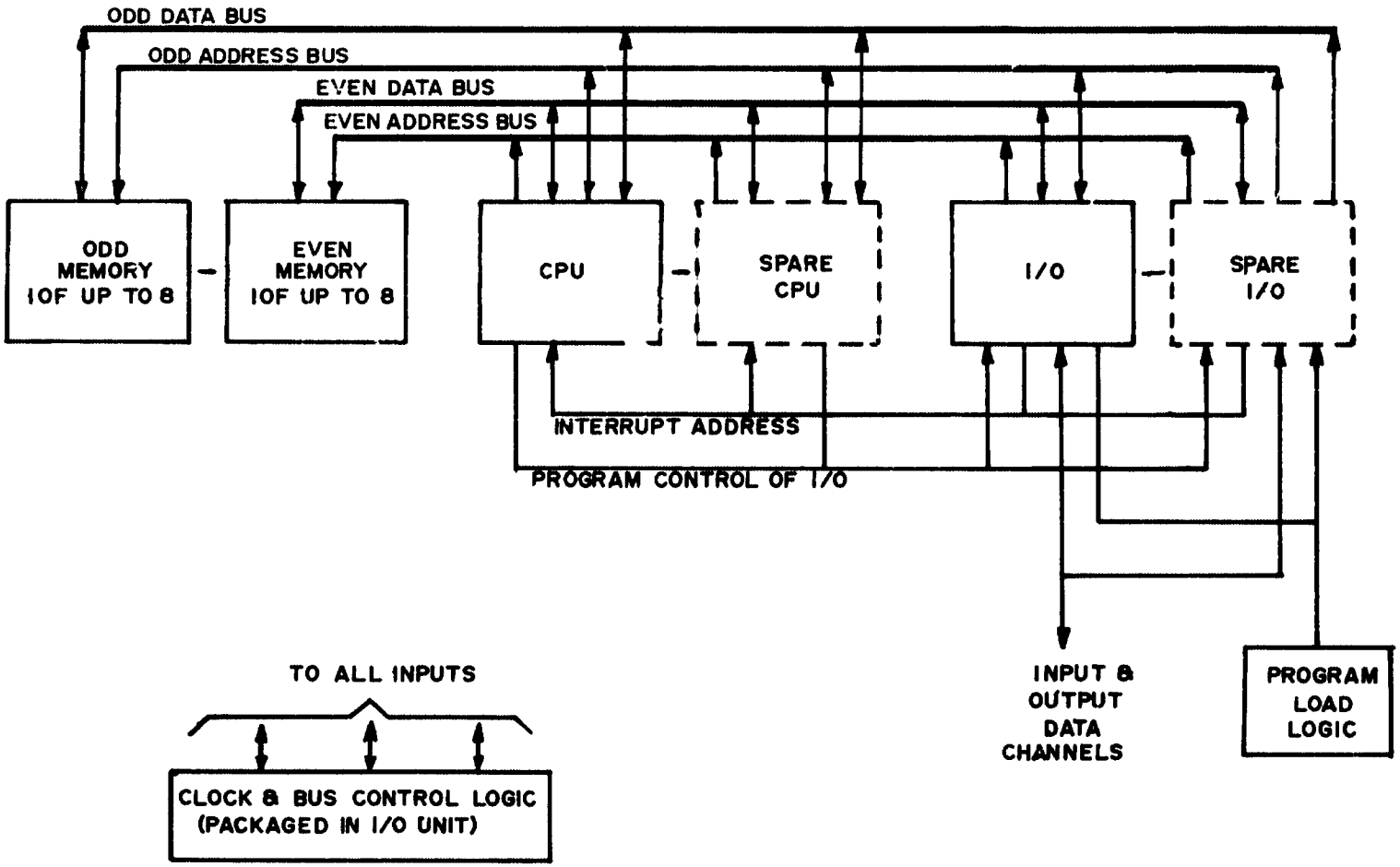


Figure 1. On-Board Processor, Functional Block Diagram

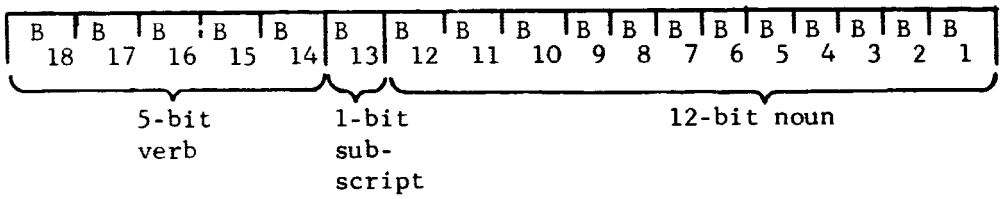
system, improves the probability of a CPU surviving for two years by a factor of 2 to 1. This redundancy configuration would require an overall increase in circuit count of approximately 50 percent.

CENTRAL PROCESSOR UNIT

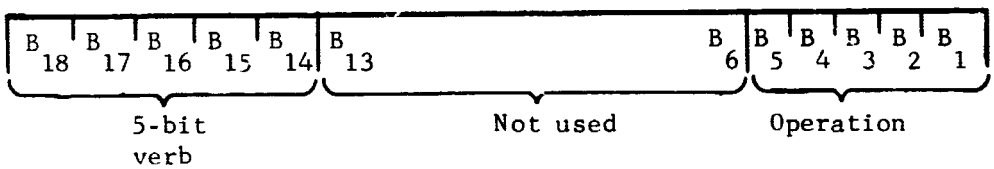
The central processor of the OBP employs a fully parallel adder and parallel data transfers between registers and at the CPU-I/O interface. Data words and instructions are 18 bits in length with negative numbers being represented in two's complement form. Addressable hardware registers include an 18-bit subscript register, an 18-bit accumulator, an 18-bit extension of the accumulator, a 4-bit memory page register, an 18-bit storage limit register which specifies read-only sections of memory, and a 6-bit scale register which represents the location of the binary point in fixed point data words. Figure 2 is a block diagram of the CPU.

Instructions are formatted as follows:

MEMORY ACCESS



NON-MEMORY ACCESS



At this stage of development, there are 50 instructions, 30 of which require an operand fetch. The other 20 instructions have a minor operation code in the address field of the instruction word. With 12 bits of address available, 4,096 memory words are directly addressable. Memory size as large as 65,536 words requires a 4-bit page register which can be loaded and stored under program control and which is appended as the four high-order bits to the 12-bit address field to form a full 16-bit address. If the subscript bit is set, the low order 16-bits of the subscript register are added to the address to form an effective address. All transfers are indirect, whereas all operand fetches are direct.

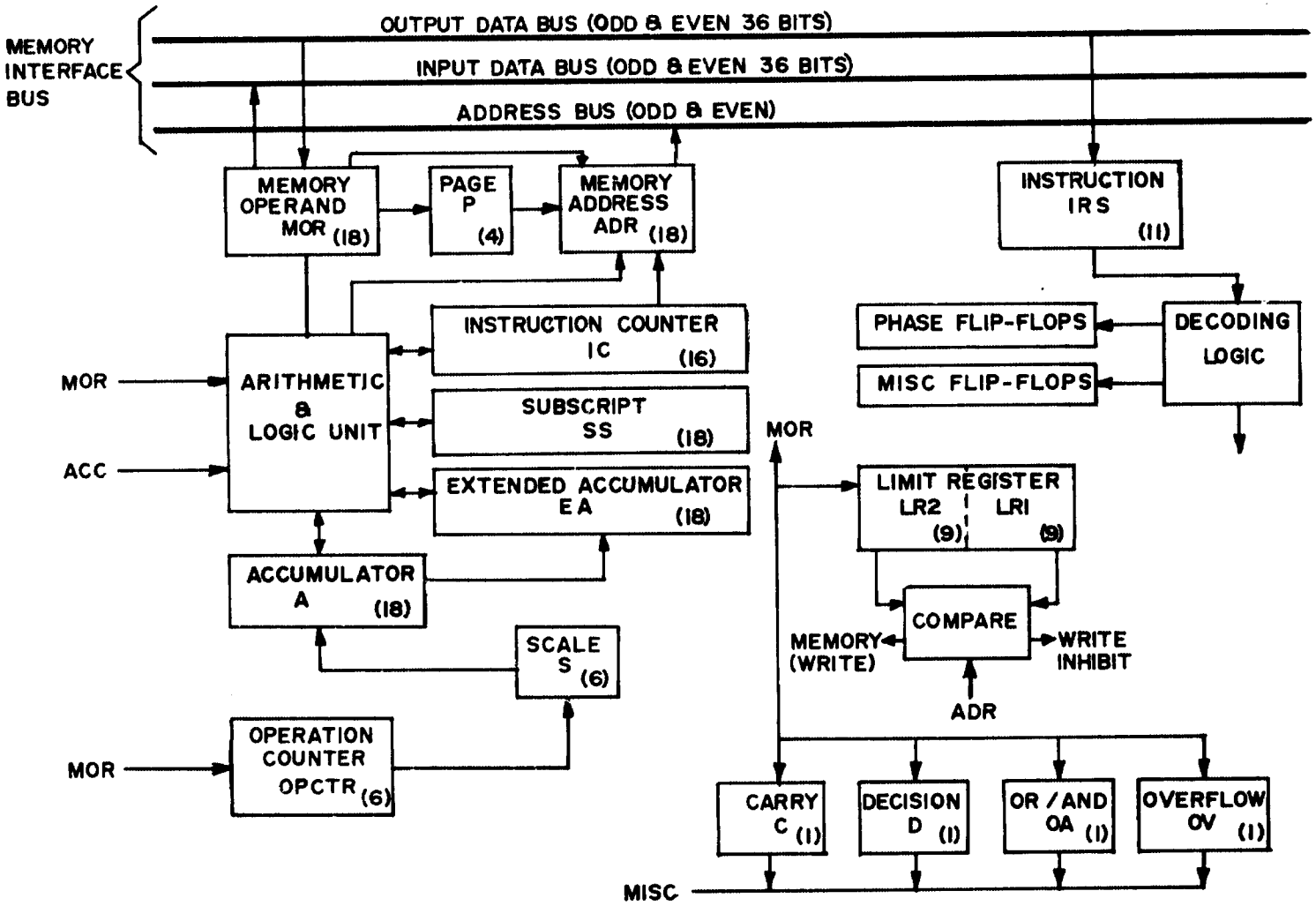


Figure 2. Central Processor Unit, Block Diagram

To prevent one user from destroying another user's program or data, the storage limit register has been employed. This register reserves a block of memory in which the operating program may write. These blocks are in increments of 128 words. To provide this feature, the 18-bit register is partitioned into two 9-bit fields. The first field specifies the most significant 9 bits of the first address that may be written. The second 9 bit field defines the most significant 9 bits of the last address that may be written.

The CPU contains a 6-bit scale register which can be loaded and stored under program control. This register is used to maintain the binary point associated with numerical quantities as a program-controlled position for multiply and divide operations, thus freeing the programmer from this chore without sacrificing accuracy as would be the case with floating point operations. The processor also employs an addressable 18-bit extended accumulator, EA, which contains the remainder following a division (the quotient is contained in the accumulator) or the low-order 17 bits, plus sign, of a 34-bit product after a multiply operation. The 36-bit accumulator/EA is automatically shifted before a divide and after a multiply, with the shift length determined by the contents of the scale register. For most operations, the programmer will be concerned with the accumulator only; however, for those instances where direct control of the EA is desirable, e.g., double precision operations, instructions are available which allow extended accumulator loading from and storing into memory locations directly. For the most part, a programmer can view the machine as having only one 18-bit arithmetic register. Standard fractional arithmetic is possible by setting the scale register to zero and standard integer arithmetic is accomplished by making the scale register equal to 17.

As shown in Figure 2, there are four addressable one-bit registers - carry, overflow, decision (D), and OR/AND (O/A) - in the CPU. The carry register is set by a carry out of bit 17 of the adder for the instructions plus, minus, and negated. If there is no carry out of bit 17 for these instructions, the carry bit will be reset. The overflow register is set during all arithmetic instructions when the result cannot be contained in the accumulator and during arithmetic shift left operations if the sign bit changes state. The overflow register can be reset only by the two instructions RESET OVERFLOW, and IF OVERFLOW. The D register is conditionally set or reset on test instructions depending on the state of the O/A register. If the O/A register is in the reset or OR state, the results of the test instructions will be ORed into the decision register. If, however, the O/A register is in the set or AND state, the results of the test instructions will be ANDed into the decision register with the previous state of the D register. The O/A register can be reset and set with the OR and AND instructions, respectively. The D register is conditionally set and reset with a reset D or a conditional jump, THEN GO TO, instruction.

Table 1 lists the functional registers.

Table 1

CENTRAL PROCESSOR UNIT FUNCTIONAL REGISTERS

Register	Symbol	Length (bits)
Memory operand	MOR	18
Memory address	MA	16
Instruction	IR	18
Instruction counter	IC	16
Subscript (index)	SS	18
Storage limit	SL	18
Page	P	4
Accumulator	A	18
Extended accumulator	EA	18
Scale	SC	6
Overflow	OV	1
Carry	C	1
Decision	D	1
Or/and	O/A	1

GRAMMATICAL MACHINE LANGUAGE

Machine Language Features

The grammatically structured machine language of the OBP was originally conceived by T. P. Gorman of the Telemetry Computation Branch, GSFC. A support software system has been developed which allows a

programmer to use either the existing English verbs, or define his own set of mnemonics. The machine language features are:

- Rules of programming are the rules of an English-like grammer.
- Vocabulary is restricted to the use of a limited number of phrases which either connect nouns or stand alone.
- Phrases correspond one for one with machine instructions.
- Nouns are the names of data quantities or the names of operation locations used in the course of a program.
- Numbers, octal or decimal, may be used directly to represent numeric quantities.
- All transfers are indirect.
- User may insert punctuation as desired to make text readable.
- Programs are written in fixed format or free form in which additional comments may be entered parenthetically within the text, thereby making the program self-documenting.
- A memory-protect feature prevents modification of instructions. Only nouns are modifiable. Subroutine return addresses are stored as though they were nouns. Therefore, once a program is loaded into memory, only data quantities will change and debugging is considerably simplified.

Instruction Set Description

This list of machine operations presents a detailed description of the execution for each instruction. The English name for the operation is called a verb to reflect the English-like language used to program the OBP. The memory access operations are denoted by noun following the operation name. This indicates the programmer must specify the name of an operand for this verb.

The instruction format and octal representation are shown in the box to the right of the programming language specification. The bit pattern of instruction words is shown on page 6.

The second instruction format, if given, is the octal representation that is produced when the configuration verb SUBSCRIPTED noun is used in the programming language.

For register definition refer to Table 1, page 9.

Load and Store Instructions

LET noun, IF noun	2 0	n o u n
LET SUBSCRIPTED noun, IF SUBSCRIPTED noun	2 1	n o u n

The content of storage at the effective address is placed in the accumulator.

LET LOCATION OF noun	4 0	n o u n
LET LOCATION OF SUBSCRIPTED noun	4 1	n o u n

The effective address is placed in the accumulator.

YIELD noun, REPLACE noun	6 0	n o u n
YIELD SUBSCRIPTED noun, REPLACE SUBSCRIPTED noun	6 1	n o u n

The content of the accumulator is stored at the effective address unless that address is protected by the storage limit registers. If storage is protected, no write into memory occurs.

SET EXTENSION WITH noun	5 2	n o u n
SET EXTENSION WITH SUBSCRIPTED noun	5 3	n o u n

The content of storage at the effective address is placed in the extended accumulator.

SAVE EXTENSION IN noun	1 0	n o u n
SAVE EXTENSION IN SUBSCRIPTED noun	1 1	n o u n

The content of the extended accumulator is stored at the effective address unless that address is protected by the storage limit registers. If storage is protected, no write into memory occurs.

Arithmetic Instructions

PLUS noun	0 4 | n o u n
PLUS SUBSCRIPTED noun	0 5 | n o u n

The content of storage at the effective address is added to the content of the accumulator and the sum is retained in the accumulator. If a carry occurs at the input of the 18th stage of the two's complement adder, then the carry register is set to 1. Otherwise, the carry register is reset to 0. Overflow can occur when two numbers of the same sign are added. Overflow causes the 18th bit of the sum to remain in the sign position and the overflow register to be set to 1.

MINUS noun	2 4 | n o u n
MINUS SUBSCRIPTED noun	2 5 | n o u n

The content of storage at the effective address is subtracted from the content of the accumulator and the difference is retained in the accumulator. Subtraction is performed by using the one's complement of the content of storage and adding it to the accumulator with a carry forced into the low-order stage of the adder. If a carry occurs at the input of the 18th stage of the two's complement adder, then the carry register is set to 1. Otherwise, the carry register is reset to 0. Overflow can occur when two numbers of different sign are subtracted. Overflow causes the 18th bit of the difference to remain in the sign position and the overflow register to be set to 1.

TIMES noun	4 4 | n o u n
TIMES SUBSCRIPTED noun	4 5 | n o u n

The content of storage at the effective address is multiplied by the content of the accumulator. The high-order 17 bits and sign of the product are retained in the accumulator. The low-order 17 bits and sign of the product are retained in the extended accumulator. The double length product is automatically scaled by arithmetically shifting the accumulator and the 17 bits of the product in the extended accumulator the number of bit positions indicated by the content of the scale register. The sign bit of the extended accumulator is not shifted. If the content of the scale register is negative,

then the shift is right and the content of the sign fills positions vacated on the left so that no overflow is possible. If the content of the scale register is positive, then the shift is to the left, with zeros filling positions vacated on the right. The overflow register is set to 1 if the sign bit of the accumulator is changed during the shift.

OVER noun, DIVIDED BY noun

6 4	n o u n

OVER SUBSCRIPTED noun, DIVIDED BY SUBSCRIPTED noun

6 5	n o u n

The content of the accumulator and extended accumulator are automatically scaled by shifting them the number of bit positions indicated by the content of the scale register. The sign of the extended accumulator is ignored and not shifted. If the content of the scale register is negative, then the shift is left with zeros filling positions vacated on the right and if the sign bit of the accumulator is changed during the shift, the overflow register is set to 1. If the content of the scale register is positive, then the shift is to the right and the content of the sign fills positions vacated on the left so that overflow is impossible. The scaled accumulator and extended accumulator form the dividend that is divided by the content of storage at the effective address. The overflow register is set if the content of the accumulator is greater than or equal to the content of storage. The signed quotient is retained in the accumulator and the signed remainder is retained in the extended accumulator. The remainder has the same sign as the dividend and has a magnitude less than the divisor.

PLUS CARRY

0 0	0 0	0 6

The content of the carry register is added to the content of the accumulator and the sum is retained in the accumulator. If a carry occurs at the input of the 18th bit of the two's complement adder, then the carry register is set to 1. Otherwise, the carry register is reset to 0. Overflow can occur and will cause the 18th bit of the sum to remain in the sign position and the overflow register to be set to 1.

NEGATED

0 0	0 0	0 4

The content of the accumulator is replaced by its two's complement. Negating all zeros yields a result of zero and sets the carry register to 1. Negating the number that has

zeros in all bit positions except the sign yields the same number as a result and sets both the carry register and the overflow register to 1. Other than these two special cases, the carry register is reset to 0.

Logic Instructions

ANDED WITH noun	3 0	n o u n
ANDED WITH SUBSCRIPTED noun	3 1	n o u n

The content of storage at the effective address is ANDed with the content of the accumulator. The result is retained in the accumulator. The 18 bits of the result are computed independently with a 1 occurring in a bit position of the result only if the accumulator and storage both contained a 1 in that bit position.

ORED WITH noun	5 0	n o u n
ORED WITH SUBSCRIPTED noun	5 1	n o u n

The content of storage at the effective address is ORed with the content of the accumulator. The result is retained in the accumulator. The 18 bits of the result are computed independently with a 1 occurring in a bit position of the result if either the accumulator or storage contained a 1 in that bit position.

EORED WITH noun, EXCLUSIVELY ORED WITH noun	7 0	n o u n
EORED WITH SUBSCRIPTED noun, EXCLUSIVELY ORED WITH SUBSCRIPTED noun	7 1	n o u n

The content of storage at the effective address is exclusively ORed with the content of the accumulator. The result is retained in the accumulator. The 18 bits of the result are computed independently with a 1 occurring in a bit position of the result if either the accumulator or storage, but not both, contain a 1 in that bit position.

COMPLEMENTED

0 0	0 0	1 0

The content of the accumulator is complemented and the result is retained in the accumulator. The 18 bits of the result are computed independently with a 1 occurring in a bit position of the result only if the accumulator contained a 0 in that position.

Bit Manipulation Instructions

SHIFTED BY noun

SHIFTED BY SUBSCRIPTED noun

1 4	n o u n
1 5	n o u n

The low-order 6 bits of the content of storage at the effective address is used as a two's complement shift count. If the count is negative, then the accumulator is shifted right the number of positions specified by the count, with the content of the accumulator sign replacing vacated positions on the left. If the count is positive, then the accumulator is shifted left the number of positions specified by the count with zeros filling vacated positions on the right. The overflow register is set to 1 if the sign bit of the accumulator is changed during the shift.

DOUBLE SHIFTED BY noun

DOUBLE SHIFTED BY SUBSCRIPTED noun

3 6	n o u n
3 7	n o u n

The low-order 6 bits of the content of storage at the effective address is used as a two's complement shift count. The accumulator and the extended accumulator are shifted together. The extended accumulator is to the right of the accumulator and its sign bit is not shifted. If the count is negative, then the accumulators are shifted right the number of positions specified by the count with the content of the accumulator sign replacing vacated positions on the left. If the count is positive, then the accumulators are shifted left the number of positions specified by the count with zeros filling vacated positions on the right. The overflow register is set to 1 if the sign bit of the accumulator is changed during the shift.

CYCLED BY noun

CYCLED BY SUBSCRIPTED noun

3 4	n o u n
3 5	n o u n

The low-order 6 bits of the content of storage at the effective address is used as a two's complement shift count. If the count is negative, then the content of the accumulator is shifted cyclically right the number of positions specified by the count, with bits leaving the low-order position entering the sign position. If the count is positive, then the content of the accumulator is shifted left the number of positions specified by the count with bits leaving the sign position entering the low-order position.

DOUBLE CYCLED BY noun

DOUBLE CYCLED BY SUBSCRIPTED noun

5 6	n o u n
5 7	n o u n

The low-order 6 bits of the content of storage at the effective address is used as a two's complement shift count. If the count is negative, then the content of the accumulator and extended accumulator is shifted cyclically right the number of positions specified by the count with bits leaving the low-order position of the extended accumulator entering the sign of the accumulator and bits leaving the low-order position of the accumulator entering the sign of the extended accumulator. If the count is positive, then the content of the accumulator and extended accumulator is shifted left the number of positions specified by the count with bits leaving the sign of the extended accumulator entering the low-order position of the accumulator and bits leaving the sign position of the accumulator entering the low-order position of the extended accumulator.

NORMALIZED

0 0	0 0	1 4

The content of the accumulator and extended accumulator is shifted left until the 17th and 18th bits of the accumulator are different. The sign bit of the extended accumulator is not shifted. Bits leaving the 17th bit of the extended accumulator enter the low-order position of the accumulator. Zeros fill the positions vacated on the right. A count of the number of positions shifted is retained as a 6-bit positive number in the scale register. If the content of the accumulator and positions 1 through 17 of the extended accumulator are zero, then the scale register is set to 0.

CLOSE EXTENSION WITH DECISION

0 0	0 0	1 3

The content of the accumulator and extended accumulator is shifted left one position. The sign of the extended accumulator is not shifted and the vacated low-order position of the extended accumulator is filled with the content of the decision register. The overflow register is not altered.

REVERSE BITS OF THE ACCUMULATOR

0 0	0 0	2 2

The contents of the accumulator are reversed in order. The $(19-n)^{th}$ and n^{th} bits are exchanges (for n = 1,9).

Subscript Instructions

USE SUBSCRIPT noun

5 4	n o u n

USE SUBSCRIPT SUBSCRIPTED noun

5 5	n o u n

The content of storage at the effective address is placed in the subscript register.

SAVE SUBSCRIPT IN noun

7 4	n o u n

SAVE SUBSCRIPT IN SUBSCRIPTED noun

7 5	n o u n

The content of the subscript register is stored at the effective address unless that address is protected by the storage limit registers. If storage is protected, no write into memory occurs.

STEP SUBSCRIPT BY noun

0 2	n o u n

STEP SUBSCRIPT BY SUBSCRIPTED noun

0 3	n o u n

The content of storage at the effective address is added to the content of the subscript register. The 18-bit result of the two's complement addition is retained in the subscript register.

Transfer Instructions

THEN GO TO noun

THEN GO TO SUBSCRIPTED noun

4 2	n o u n
4 3	n o u n

If the content of the decision register is zero, then the next sequential instruction is executed. If the content of the decision register is one, then the content of storage at the effective address is placed in the instruction counter and execution proceeds from the address specified by the instruction counter. The decision register and OR/AND register are reset to zero.

GO TO noun, RETURN FROM noun

GO TO SUBSCRIPTED noun, RETURN FROM SUBSCRIPTED noun

6 2	n o u n
6 3	n o u n

The content of storage at the effective address is placed in the instruction counter and execution proceeds from the address specified by the instruction counter.

TRANSFORMED BY noun, PERFORM noun

TRANSFORMED BY SUBSCRIPTED noun, PERFORM SUBSCRIPTED noun

0 6	n o u n
0 7	n o u n

The content of the instruction counter plus one is stored at the effective address unless that address is protected by the storage limit registers. If storage is protected, no write into memory occurs. The content of one location greater than the effective address is placed in the instruction counter and execution proceeds from the address specified by the instruction counter.

Test Instructions

AND

0 0	0 0	1 1

The OR/AND register is set to one.

OR

0 0	0 0	1 5

The OR/AND register is set to zero.

IS LESS THAN noun

IS LESS THAN SUBSCRIPTED noun

2 6	n o u n
2 7	n o u n

If the content of the accumulator is less than the content of storage at the effective address, then the test condition is set to 1. Otherwise, it is zero. The OR/AND register being zero specifies that the test condition is to be ORed with the content of the decision register and the result is to be retained in the decision register. The OR/AND register being one specifies that the test condition is to be ANDed with the content of the decision register and the result is to be retained in the decision register.

IS EQUAL TO noun

IS EQUAL TO SUBSCRIPTED noun

4 6	n o u n
4 7	n o u n

If the content of the accumulator is equal to the content of storage at the effective address then the test condition is set to 1. Otherwise it is zero. The OR/AND register being zero specifies the test condition is to be ORed with the content of the decision register and the result is to be retained in the decision register. The OR/AND register being one specifies the test condition is to be ANDed with the content of the decision register and the result to be retained in the decision register.

IS GREATER THAN noun

IS GREATER THAN SUBSCRIPTED noun

6 6	n o u n
6 7	n o u n

If the content of the accumulator is greater than the content of storage at the effective address the test condition is set to 1. Otherwise, it is zero. The OR/AND register being zero specifies that the test condition is to be ORed with the content of the decision register and the result is to be retained in the decision register. The OR/AND register being one specifies that the test condition is to be ANDed with the content of the decision register and the result is to be retained in the decision register.

F OVERFLOW

0 0	0 0	0 1

If the content of the overflow register is one, then the test condition is set to one. Otherwise, it is zero. The overflow register is reset to zero. The OR/AND register being zero specifies that the test condition is to be ORed with the content of the decision register and the result is to be retained in the decision register. The OR/AND register being one specifies that the test condition is to be ANDed with the content of the decision register and the result is to be retained in the decision register.

IF PARITY ODD

0 0	0 0	0 5

If the number of ones in the 18-bit accumulator is odd, then the test condition is set to one. Otherwise, it is zero. The OR/AND register being zero specifies that the test condition is to be ORed with the content of the decision register and the result is to be retained in the decision register. The OR/AND register being one specifies that the test condition is to be ANDed with the content of the decision register and the result is to be retained in the decision register.

IS POSITIVE

0 0	0 0	0 3

If the sign position, bit 18, of the accumulator contains a zero, then the test condition is set to one. Otherwise, it is zero. The OR/AND register being zero specifies that the test condition is to be ORed with the content of the decision register and the result is to be retained is the decision register. The OR/AND register being one specifies that the test condition is to be ANDed with the content of the decision register and the result is to be retained in the decision register.

IS EQUAL TO ZERO

0 0	0 0	2 1

If the value of the contents of the accumulator is equal to zero, then the test condition is set to one. Otherwise, it is zero. The OR/AND register being zero specifies that the test condition is to be ORed with the content of the decision register and the result is to be retained in the decision register. The OR/AND register being one specifies that the test condition is to be ANDed with the content of the decision register and the result is to be retained in the decision register.

IF SUBSCRIPT IS NOT GREATER THAN noun

IF SUBSCRIPT IS NOT GREATER THAN SUBSCRIPTED noun

2 2	n o u n
2 3	n o u n

If the content of the subscript register is less than or equal to the content of storage at the effective address, then the test condition is set to one. Otherwise, it is zero. The OR/AND register being zero specifies that the test condition is to be ORed with the content of the decision register and the result is to be retained in the decision register. The OR/AND register being one specifies that the test condition is to be ANDed with the content of the decision register and the result is to be retained in the decision register.

IS FALSE

0 0	0 0	1 7

The decision register is complemented.

Control Instructions

PASS

0 0	0 0	0 2

No operation is performed other than the automatic incrementing of the instruction counter.

HALT

0 0	0 0	0 0

The processor stops indefinitely. An initiate signal must be supplied from an external source to start the Processor.

EXECUTE noun

EXECUTE SUBSCRIPTED noun

1 2	n o u n
1 3	n o u n

The content of storage at the effective address is used as the address of the instruction to be executed. The instruction counter is incremented by one unless it is changed by the execution of a transfer-type instruction. If the machine attempts to execute an EXECUTE noun, the program proceeds with no operation being performed.

Miscellaneous Register Instructions

SET SCALE WITH noun

SET SCALE WITH SUBSCRIPTED noun

3 2	n o u n
3 3	n o u n

The low-order 6 bits of the content of storage at the effective address is placed in the scale register.

LET SCALE

0 0	0 0	2 0

The content of the scale register is placed in the low-order 6 bits of the accumulator. The high-order 12 bits of the accumulator are set to zero.

SET PAGE

0 0	0 0	1 2

The content of bits 13 through 16 of the accumulator are placed in the page register.

RESET OVERFLOW

0 0	0 0	0 7

The content of the overflow register is set to zero.

RESET DECISION

0 0	0 0	2 3

The contents of the decision register is set to zero.

EXIT

0 0	0 0	1 6

This instruction initiates interrupt number 16 and uses locations octal 200 through 207. Upon completion, execution proceeds normally using the new value in the instruction counter.

RESUME FROM noun

RESUME FROM SUBSCRIPTED noun

7 2	n o u n
7 3	n o u n

The content of storage at the effective address is used as the starting address of an interrupt storage area. This instruction reloads the registers that were saved at the occurrence of an interrupt. Upon completion, execution proceeds normally

using the new value in the instruction counter.

Input/Output Instructions

CONNECT TO noun

CONNECT TO SUBSCRIPTED noun

1 6	n o u n
1 7	n o u n

The content of storage at the effective address is sent to the I/O unit as a command. This word must have a zero in bits 17 and 18 to denote the establishment of a cycle steal channel. Bits 1 through 16 specify the data channel and block length. The content of the accumulator is stored at location seven. The content of location seven is then output as a starting memory address to the I/O unit.

LET FUNCTION TO noun

LET FUNCTION TO SUBSCRIPTED noun

1 6	n o u n
1 7	n o u n

The content of storage at the effective address is sent to the I/O unit as a command. This word must have a zero in bit 18 and a one in bit 17 to denote that bits 1 through 16 are a function code.

OUTPUT TO noun

OUTPUT TO SUBSCRIPTED noun

1 6	n o u n
1 7	n o u n

The content of storage at the effective address is sent to the I/O unit as a command. This word must have a one in bit 18 and a zero in bit 17. Bits 1 through 16 denote the data channel for an output device. The content of the accumulator is stored at location seven is then output as data to the I/O unit.

LET INPUT FROM noun

LET INPUT FROM SUBSCRIPTED noun

1 6	n o u n
1 7	n o u n

The content of storage at the effective address is sent to the I/O unit as a command. This word must have a one in bits 17 and 18. Bits 1 through 16 denote the data channel for an input device. The I/O unit stores one word of data at location seven. The content of location seven is then placed in the accumulator.

INPUT/OUTPUT UNIT

Design objectives require that the On-Board Processor be applicable to a variety of scientific spacecraft programs. To be successful in this approach, a general and versatile technique for data transfer and interrupt processing was developed. The intention is to tailor the computer's I/O scheme to accommodate a particular mission's individual and perhaps unique requirements, with no impact on the CPU and memory unit design.

With this objective in mind, the system design has been established to provide:

- Data transfer external to program execution (called the cycle steal mode)
- Data transfer completely under program control

In the first method, cycle steal control, the I/O time shares the memory with the CPU, allowing continuous data input or output without interrupting CPU processing. In cycle steal, data can be transferred at rates up to 4×10^5 words per second. The second method, program-controlled data transfer, may be requested in two ways. The faster is to interrupt the program. Whenever an interrupt occurs, the processor (upon completion of the current instruction) leaves the program it is processing, goes to the routine associated with that particular interrupt, and there executes the appropriate input/output instructions. After the routine has been completed the Processor returns to the original program. A slower type of program-controlled data transfer is one in which the Processor periodically scans a register to determine if an input/output request has occurred. If so, the Processor acts on the particular request according to its priority. This type of data transfer is feasible when instruction execution rates are fast compared to the input data rates.

During design of the I/O subsystem, for the engineering model, it became evident that a significant section of the I/O could remain unchanged for a variety of applications. This section of the system was defined, and is referred to as the standard I/O unit. A Functional Block Diagram is shown in Figure 3. Sections of the system which will change with the application are special data sources and receivers, such as: A to D converters, multiplexers, D to A converters, counters and serial to parallel registers. These are referred to as peripheral devices.

Cycle Steal Operation

As shown in Figure 3, the standard I/O includes two cycle steal control channels (A and B) which can be initialized by command from the ground or by the CPU program. These control channels have been implemented to provide addressing and read/write control to the memory bus for

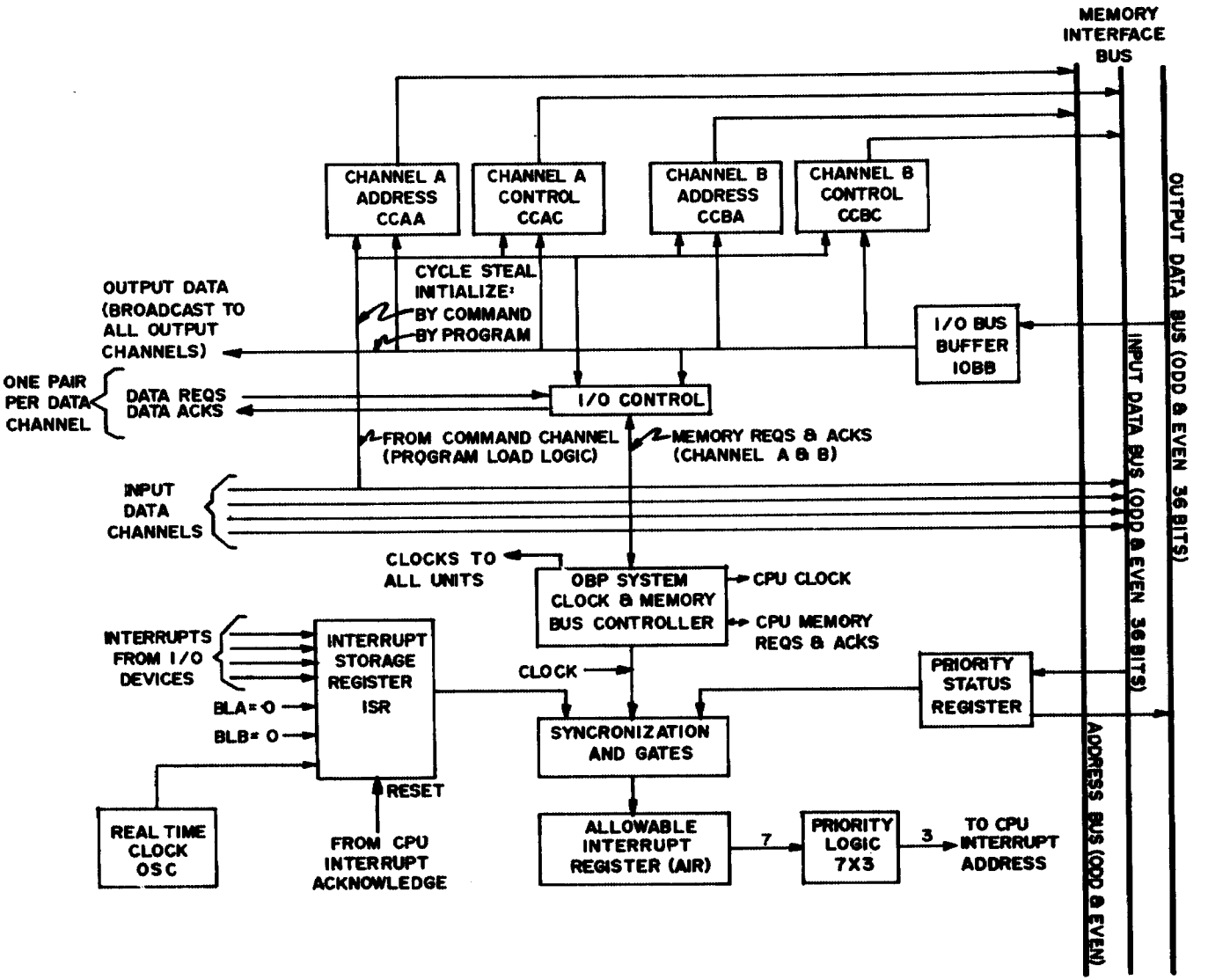


Figure 3. I/O Unit, Functional Block Diagram

any data channel assigned to operate in the cycle steal mode. Two control channels were included to furnish redundant addressing circuits for the command system (program load channel) and provide a flexible direct memory access capability which can be time shared among the other data channels. The two control channels are identical and separate. Each contain a 16-bit address register and a 14-bit control register which stores the block length (12 bits) and data channel assignment (2 bits). When a given control channel is in operation it will be maintained at a device-dependent rate by interfacing the device and memory using the well known Request/Acknowledge technique of synchronization. The individual input data channels are tied directly to the input data bus through gated driver circuits. Output data channels receive data from the output bus buffer which serves as a broadcast register to all devices. Each input or output device receives a unique acknowledge control line when being serviced with a data transfer. Logic levels at the input/output interface are compatible with Low Power DTL 9040 series circuits.

Program Control Operation

The CPU instruction repertoire includes an input/output instruction (order code 16 or 17) which either allows a data output, a data input, a function output, or activates control channels A or B (refer to Input/Output instruction Table 2). One of these four operations will be executed as specified by the contents of location M, the effective address in the I/O instruction. The format for this word may vary with the application, however, the format which has been generally defined for the engineering model implementation is detailed in Table 2. The manner in which the CPU accumulator (designated A) will use the instruction is also dependent on which of the four functions is executed.

Interrupt Operation

At the completion of each instruction, the Processor senses if any of the 15 external interrupts appeared during the execution of the previous instruction. The order of hardware priority runs from channel 1 as the highest to channel n as the lowest priority. The hardware priority applies only to prevent conflict when several interrupts allowed by priority status are waiting to be processed at the end of an instruction. The priority status word provides programming control of the order in which interrupts will be processed.

The priority status register, as shown in Figure 3, is loaded by a data transfer from the memory to the input/output unit in which bits 1 through n correspond to the n interrupt levels. The programmer determines the bit pattern contained in the interrupt location and thereby sets the priority status.

Table 2

INPUT/OUTPUT FORMAT

CONNECT TO M

	18	17	16	15	14	13	12---------1
---M---	0	0	CC	X	Device #		Block Length
---A---	18	17	16---------1				
	X	CC	Starting ADD				

CC = 0 Channel A is selected
CC = 1 Channel B is selected

LET FUNCTION TO M

	18	17	16---------1
---M---	0	1	Bits to be defined
---A---	18		1
	X		X

OUTPUT TO M

	18	17	16----13	12----------1
---M---	1	0	Device #	X X
---A---	18			1
	Any Data to be output			

See note

LET INPUT FROM M

	18	17	16 13	12----------1
---M---	1	1	Device #	X X
---A---	18			1
	Holds incoming word			

See note

NOTE: Device #'s 0000_2 to 0011_2 are identical to Device #'s 00_2 to 11_2 for the "CONNECT TO" instruction.

When an interrupt arrives at the I/O unit on one of the interrupt lines, it is compared with the corresponding bit position of the priority status register. If the current priority status has a 1 in the respective bit position for that interrupt, then the interrupt is temporarily locked out and cannot get to the Processor. As shown in the block diagram, the interrupt storage register holds the interrupt so that if, at some later time, the priority status is changed and a 0 appears in the corresponding bit position of the saved interrupt, then the interrupt will be honored and allowed to pass through synchronization gates to the allowable interrupt register. Since several allowable interrupts may exist simultaneously, the priority logic allows only one to pass to the CPU. This logic also converts the interrupt number to a binary code. When the Processor receives the interrupt, an acknowledge signal is sent back to the I/O unit, resetting the flip-flop so that another interrupt may be saved. Otherwise, an interrupt arriving at the I/O unit on a channel that is currently holding a previous interrupt will be lost. The priority status is altered when an interrupt, an EXIT instruction, or a RESUME FROM instruction is processed by the CPU.

When an interrupt is honored, say interrupt n, fixed memory locations as illustrated by Table 3 are used to store contents of unaddressable registers. The following registers are stored into and loaded from the indicated interrupt locations.

Interrupt priority status → location 8 x n (bits 1 to 16)

Scale Register → location 8 x n + 1 (bits 1 to 6)

Carry Register → location 8 x n + 1 (bit 7)

Overflow Register → location 8 x n + 1 (bit 8)

OR/AND Register → location 8 x n + 1 (bit 9)

Decision Register → location 8 x n + 1 (bit 10)

Page Register → location 8 x n + 1 (bits 13 to 16)

Limit Register → location 8 x n + 2 (bits 1 to 18)

Instruction Counter → location 8 x n + 3 (bits 1 to 16)

Location 8 x n + 4 (bits 1 to 16) → interrupt priority status

Location 8 x n + 5 (bits 1 to 6) → scale register

Location 8 x n + 5 (bit 7) → carry register

Location 8 x n + 5 (bit 8) → overflow register

Location 8 x n + 5 (bit 9)→OR/AND register

Location 8 x n + 5 (bit 10)→decision register

Location 8 x n + 5 (bits 13 to 16)→page register

Location 8 x n + 6 (bits 1 to 18)→limit registers

Location 8 x n + 7 (bits 1 to 16)→instruction counter, upon loading of the instruction counter, program execution proceeds using the new address.

Table 3

INTERRUPT LOCATIONS

	18 17 16	15 14 13 12 11 10	9	8	7	6 5 4 3 2 1
0		OLD INTERRUPT PRIORITY				
1		PAGE	D	OA	OV	C	SCALE
2		OLD STORAGE LIMIT REGISTER				
3		OLD INSTRUCTION COUNTER				
4		NEW INTERRUPT PRIORITY				
5		PAGE	D	OA	OV	C	SCALE
6		NEW STORAGE LIMIT REGISTER				
7		NEW INSTRUCTION COUNTER				

When the RESUME FROM noun instruction is executed, at the conclusion of interrupt processing, the following registers are restored.

Location 8 x n (bits 1 to 16)→interrupt priority status

Location 8 x n + 1 (bits 1 to 6)→scale register

Location 8 x n + 1 (bit 7)→carry register

Location 8 x n + 1 (bit 8)→overflow register

Location 8 x n + 1 (bit 9) ⟶ OR/AND register

Location 8 x n + 1 (bit 10) ⟶ decision register

Location 8 x n + 1 (bits 13 to 16) ⟶ page register

Location 8 x n + 2 (bits 1 to 18) ⟶ limit register

Location 8 x n + 3 (bits 1 to 16) ⟶ instruction counter

Upon completion, execution begins at the new value of the instruction counter.

OBP System Clock

The computer system clock generates a 1.6 MHz square wave which is divided into two 0.8 MHz timing signals which have relative phasing of 180°. Both outputs of the two-phase clock are asymmetrical pulses having a duration of 312.5 nanoseconds and a period of 1250 nanoseconds. The clock pulse duration is a function of bistable multivibrator response time, and clock repetition rate is a function of the CPU carry propagation time. A third output of the system clock is a 500 nanosecond pulse occurring at 0.8 MHz. This pulse serves as the memory system initiate signal: the 500 nanosecond duration is independent of the clock rate. The OBP clock circuit is physically packaged with the I/O portion of the system.

Memory Bus Controller

This circuit supervises the activity of the memory system. In the event that Channel A, Channel B, and the CPU are requesting memory cycles simultaneously, the controller establishes a priority of memory use so as to avoid erratic operation. The order of priority is Channel A first, Channel B second, and the CPU last. In order to prevent a short or open circuit on a channel request line from capturing all memory cycles, the logic design requires that a request must be dropped after each memory cycle and be raised again prior to honoring of a second memory cycle. Like the OBP clock, the memory bus controller is contained in the I/O package.

CIRCUITRY

Integrated circuits are used extensively in the On-Board Processor. Circuits selected for the engineering model are the Fairchild 9040 series, a low-power diode transistor logic (LPDTL) employing a +5-volt power supply. The logic is packaged in 14 lead ceramic flat packs, and

will operate at speeds up to 2.5 MHz with power dissipations of 1 mw per gate (50 percent duty cycle) and 4 mw per flip-flop. These circuit elements are readily available and have one of the best power-speed products of any logic type presently available in large quantities from stock. The 9040 series, a standard line which has been in existence over 3 years, is well established and has a good reliability history.

FLIGHT MEMORY SYSTEM

The first flight will employ a memory system composed of several core memory units connected to EVEN and ODD Data and Address buses. Each memory unit has a data capacity of 4,096 18-bit words, and occupies 132 cubic inches with dimensions of 8.0 by 4.5 by 3.65 inches.

The system is configured so that half of the units are connected to the EVEN bus set and half to the ODD bus set, as shown in Figure 1, page 5. This configuration ensures that any single failure in any bus interfacing circuit cannot cause a complete system malfunction.

The maximum memory configuration is composed of 8 units on each bus set providing a data capacity of 65,536 18-bit words, and consuming approximately 30 watts of power while operating at a rate of 4×10^5 cycles per second. In standby status power consumed is negligible. This modest power requirement for a destructive readout core implementation is attained by switching power to all units on a cycle by cycle basis. Consequently, a unit will consume power only when it is addressed. In standby the power required by each unit is low since only the logic circuitry required to detect its address is energized.

It is anticipated that a further reduction in power and volume, in future flight systems, can be realized by employing nondestructive readout plated wire memories. Significant advances are being made in the development of ruggedized plated wire systems, and they will be incorporated in the OBP as proven hardware becomes available.

PACKAGING

Engineering Model

The engineering model (refer to Figure 4) of the CPU and I/O units contain small pluggable cards (4.5 x 1.5 inches) on which a maximum of 12 flat packs can be mounted. The flat packs are stacked two high with the top one oriented 90 degrees with respect to the bottom pack. The printed circuit card contains a blade-type connector which mates with a tuning fork female contact. The printed circuit cards are general purpose and all interconnections and intraconnections are wire-wraps performed on the backboard. Wiring lists are prepared by computer and the wiring can be performed either automatically or by hand.

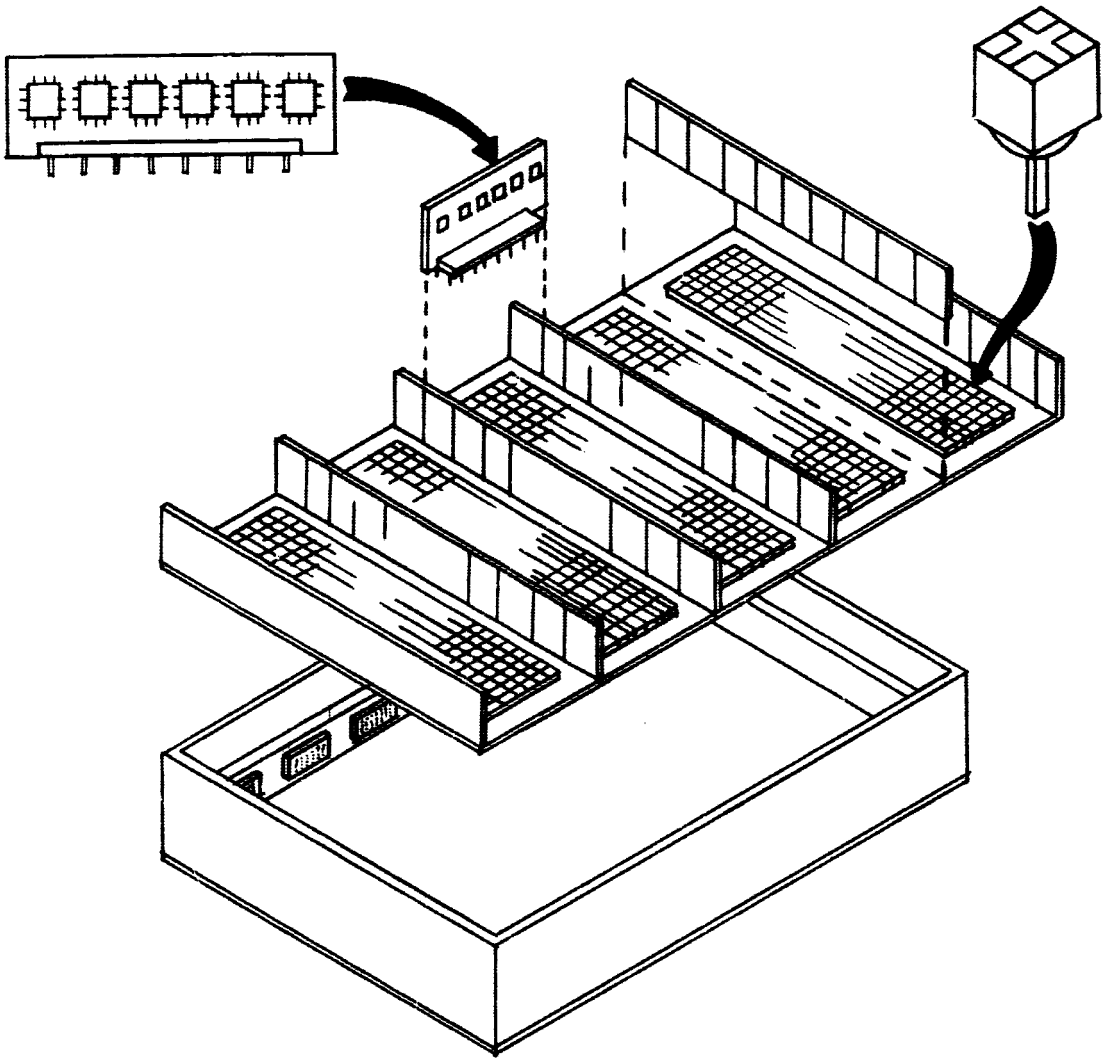


Figure 4. Engineering Model Packaging of CPU and I/O Units

A total of 138 cards are required to fabricate the CPU. The configuration is 5 rows of up to 30 cards each on 0.3 inch centers. The connector panel has slotted card guide assemblies mounted into it and is assembled into a housing with input/output connectors positioned along the back.

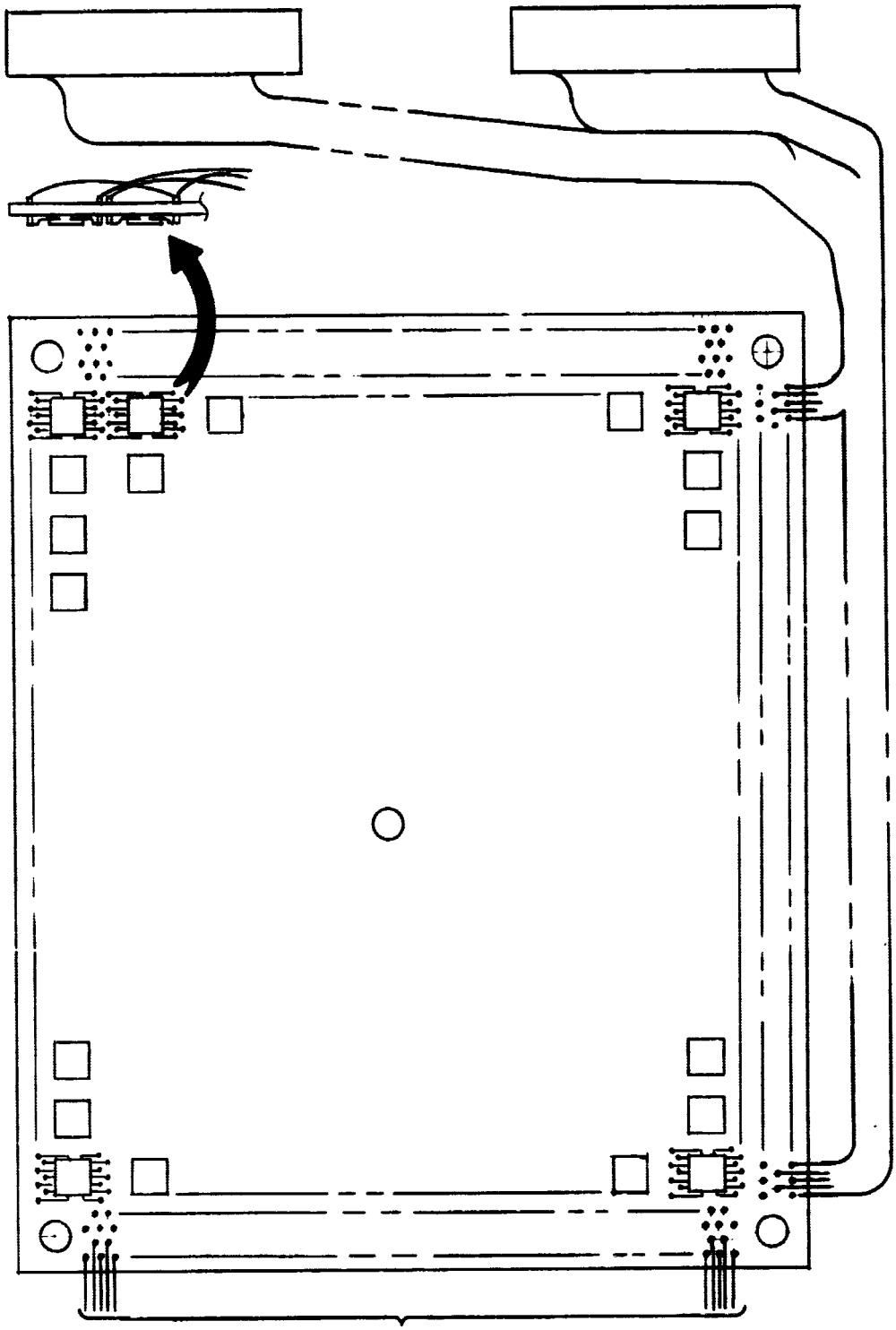


Figure 5. Flight Model Board Intraconnections

Flight Model

The proposed flight package will contain individual integrated circuit flat packs mounted flat on perforated boards (refer to Figure 5). The boards will be 6.5 x 5.5 x 0.063 inches, and will be perforated with 25 mil holes having 50 mil spacing throughout the entire board. Interconnect pins of weldable material will be inserted at selected locations on the board to provide for backboard intraconnects between flat packs. The pins will be welded to the flat pack leads; and the mounted flat packs will be intraconnected on the back of the board by a point to point weld using insulated wire and a cold electrode weld technique, or other suitable method. The completed backboard wiring will be conformally coated to prevent damage to the wire. The above technique allows for a maximum of 133 flat packs to be mounted on each board.

The CPU package will consist of approximately 1,500 flat packs mounted on 12 mother boards, and the standard I/O package will consist of approximately 1,000 flat packs mounted on 8 mother boards. Since, so many boards will be required to complete an individual unit an intermediate board called the "grandmother board," and located in the center of the stack, will be used to perform intraconnects between boards. In both units flexible flat conductor cables will be welded to one end of the mother boards, and to both ends of the "grandmother board" (refer to Figure 5). In this way boards located at the extreme ends of the stack will have their interconnecting leads pass through a single board rather than all boards located between the two boards. External connections to the boards will be provided by cabled wires from one side of each board to mounted rectangular miniature connectors on the face of the units, as shown in Figure 6. The CPU will have 10 connectors (5 are for ground checkout equipment), and the number required on the I/O unit will vary.

When all connections are completed the boards will be folded together in a book-like construction, as shown in Figure 7. Foam sheets will be used to separate the cards and dampen vibration. The entire card stack will be held in compression by means of a rigid honeycomb plate on top of the stack held down by the one center and four corner screws of the stack. The dimensions of the CPU package will be 8 by 6 by 4.5 inches, and that of the I/O will be 8 by 6 by 4 to 4.5 inches, with each unit weighing approximately six pounds. The base of the units will have four mounting feet used to secure them within the spacecraft.

The memory unit will be conventional destructive readout core (DRO) memory. Each 4K by 18 bit memory package will have dimensions of 8.0 by 4.5 by 3.65 inches, and a total weight of 6.9 pounds.

The first flight qualified OBP will be composed of one CPU, one I/O unit, a power converter, and probably four memory units. The overall package volume, will be less than 1,100 cubic inches, and weigh less than 45 pounds.

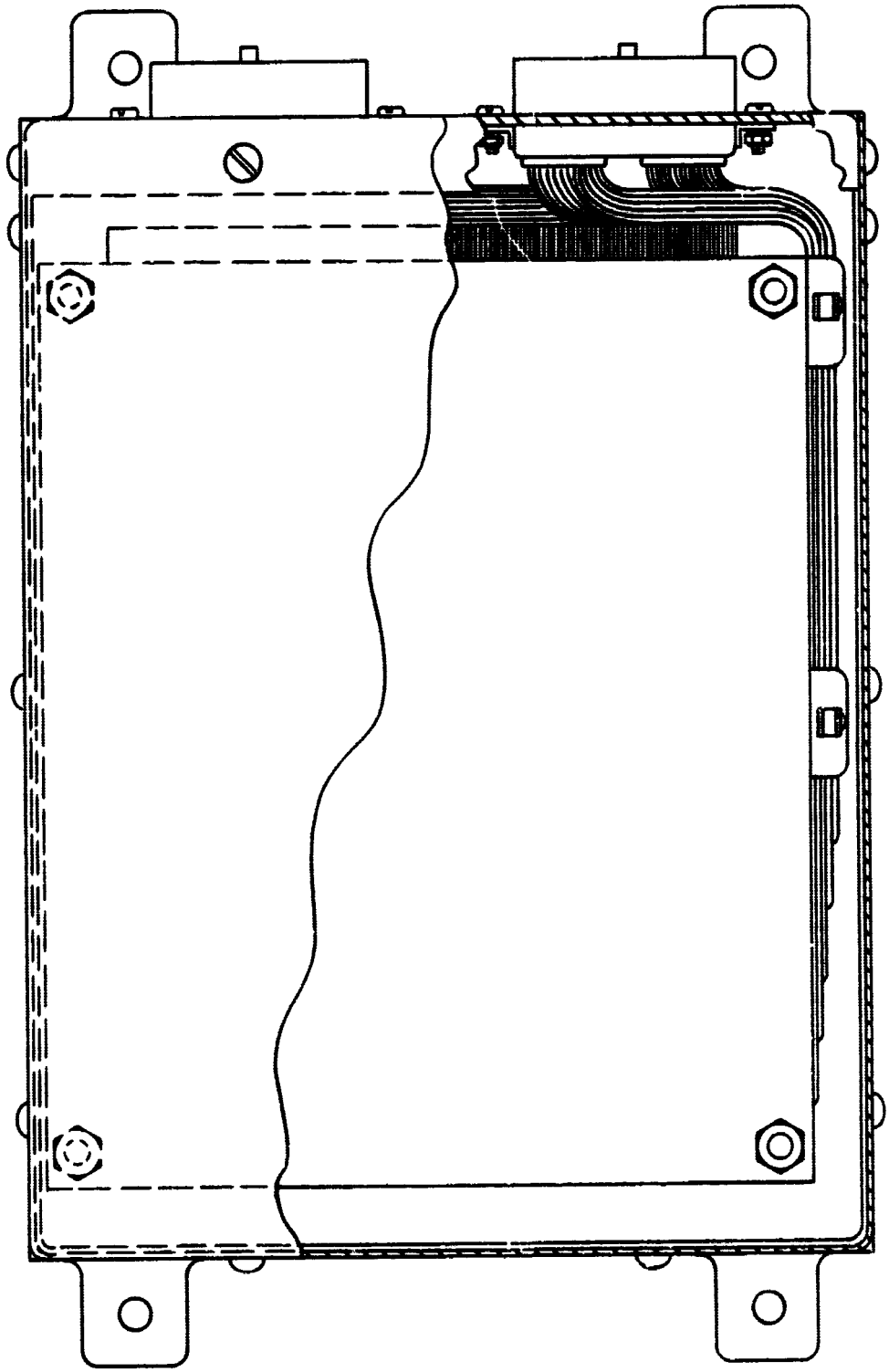


Figure 6. Flight Model Connector Cabling

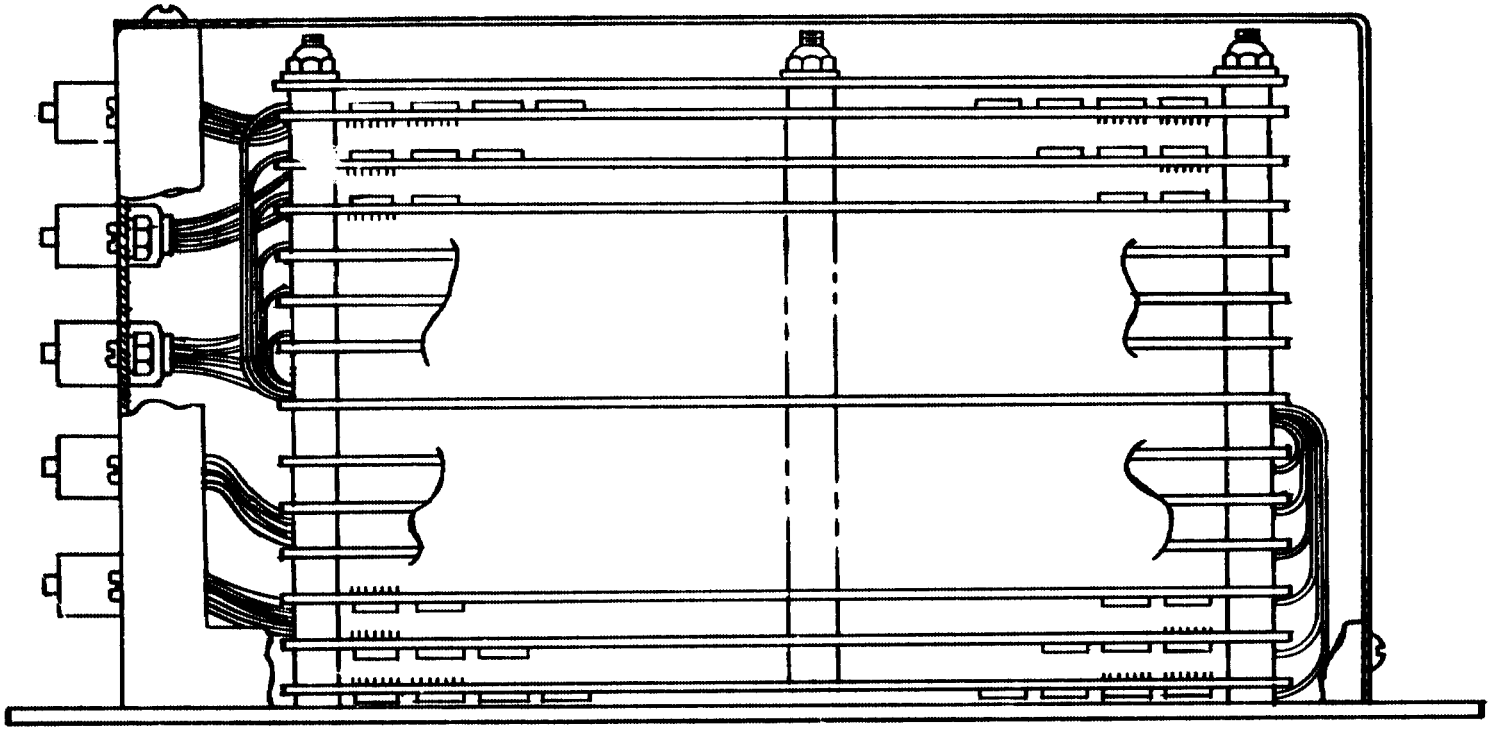


Figure 7. Flight Model Packaging of CPU and I/O Units

For future flights - within three years - plans are to develop a nondestructive readout (NDRO) plated wire memory with a projected savings in size and weight of approximately 40 percent. The use of LSI and hybrid techniques in the construction of the CPU and I/O units is being considered for future models. Information regarding configuration of future units will be documented in later reports.

APPLICATIONS

Functional Uses

Incorporating a computer aboard a spacecraft allows the concept of an integrated system to be realized. Integrated system here refers to the unifying of both spacecraft and experiment functions through a central device - an On-Board Processor. Due to its inherent flexibility, the OBP can easily be adapted to support various spacecraft. Its use could eliminate unnecessary redundancy and simplify the overall hardware requirements.

Another task which the OBP can handle is command storage and execution. This requires that the following operations be performed: command verification; error checking; command storage; command distribution; and delayed execution. Many of these operations are now performed on board by special purpose equipment. The hardware required to perform this task is inherently a part of the OBP.

Another task which the OBP can perform is control of spacecraft attitude and stabilization. In maintaining or changing the spacecraft's attitude position, the OBP would be responsible for generating the necessary commands or control signals to (re) orient the spacecraft through a closed-loop control system. The OBP can maintain continuous attitude information and thereby achieve a fast and efficient transition from one position to another, resulting in a savings in time and possibly control gas.

Spacecraft subsystem monitoring and diagnosis can also be a function of the OBP. This requires spacecraft data utilization and can be performed by what is known as limit checks. A table of acceptable limit values for subsystem parameters, such as temperatures and currents, would be compared with the measured values. If a parameter were out of limits, this information could be telemetered to the ground, or planned remedial action could be taken by the OBP. Operational subsystem checking could be performed as well. This would entail checking for the correct spacecraft response to an OBP generated command, which requires a comparison of status data with the desired commanded status. Finally, subsystem operations such as forming, formatting and coding telemetry frames, and data compression and smoothing could be handled by the OBP.

Similar functions to those mentioned above could be performed for the experiment data. Such things as experiment control - either control of sampling or adjustment of the experiment parameters - could be a function of the OBP. One major service could be correlation of experiment data with spacecraft attitude, command information, spacecraft status, time, and other experiment data.

Only a few of the possible uses of an OBP have been mentioned. Its potential usefulness to both the spacecraft and the experimenter would be difficult to exaggerate. A system which can be reconfigured by software changes has almost limitless potential.

LABORATORY CONFIGURATION

The OBP laboratory configuration has been implemented in such a way as to provide a versatile and flexible spacecraft computer simulation. To this end an SDS 920 computer has been connected to the OBP via four coaxial cables and two appropriate interface units, as shown in Figure 8. These cables carry serial data and clock to and from the OBP. This allows simulation of both a command and telemetry link. The serial data rate employed is 50,000 bits per second.

The configuration shown also provides the communication necessary for the 920 to simulate a complete spacecraft environment. The 920 provides the OBP with program loads and commands, and could produce a variety of simulated experiment data, spacecraft sensor data, momentum data, etc. Further the SDS 920 receives and responds to program dumps, simulated telemetry frames, and discrete commands from the OBP. If properly programmed the 920 system could simulate the spacecraft kinematics.

Looking again at Figure 8, it can be observed that various devices have been provided for the programmer to load programs into the OBP or to output the program results. Inputs are allowed from the card reader, paper tape reader, typewriter, or magnetic tape. The typewriter, magnetic tape, paper tape and high-speed printer can be used as output devices. A special program has been prepared for the SDS 920 to make it appear as a hardwired peripheral device for OBP use.

Data and programs can also be inputted to the OBP by the control panels on the laboratory console. These control panels, used as ground checkout equipment, display all of the various operational registers, as well as manual execution of programs and sequences. The control panels are used primarily as debugging aids and would not normally be employed during operation unless a quick visual status check was desired.

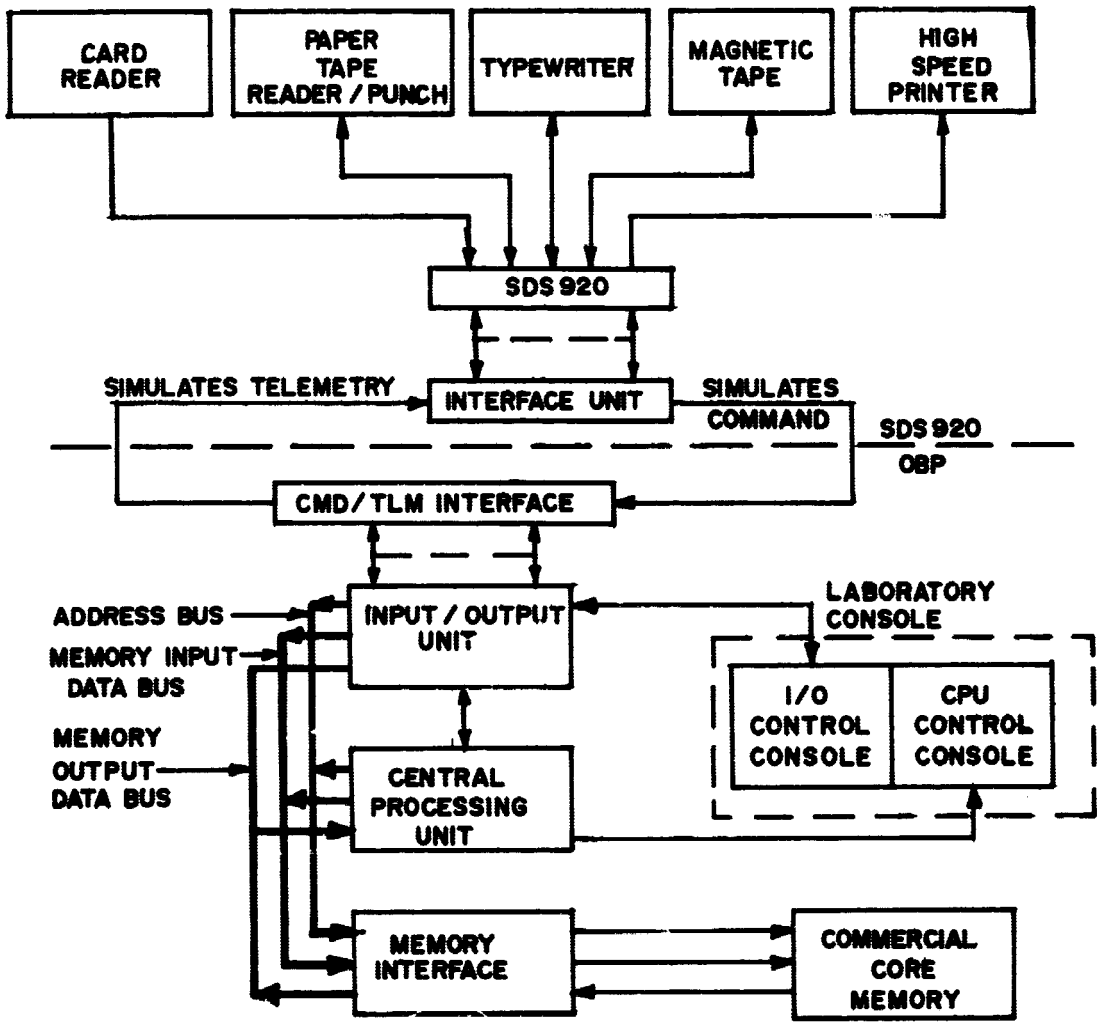


Figure 8. OBP Laboratory Configuration